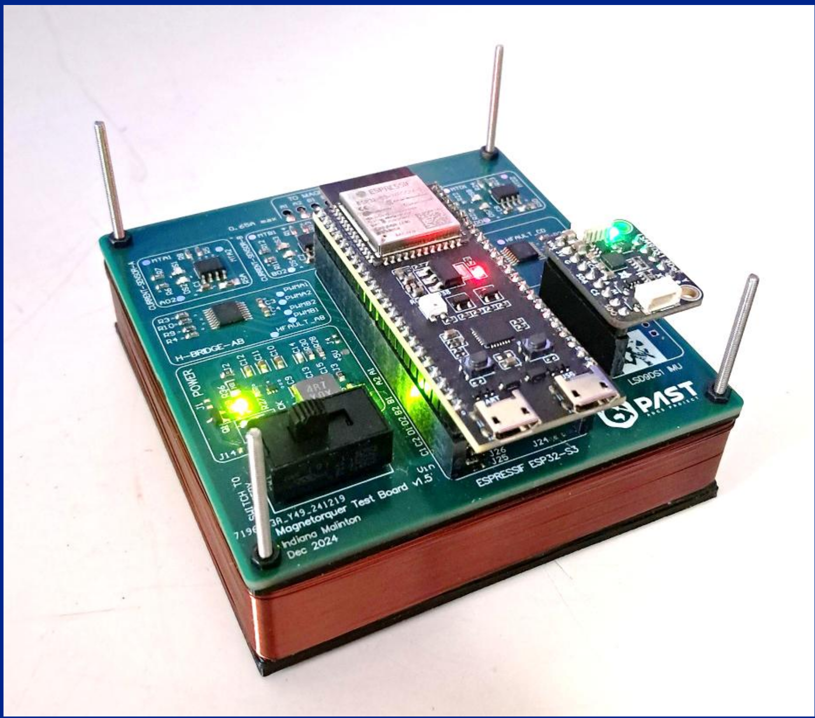


Magnetorquer Testing Module

September 2024 – January 2025

Indiana Malinton
22198356@student.curtin.edu.au | Indiana.malinton@gmail.com
Perth Aerospace Student Team (PAST)

Contents

1.0 Project Overview

1.1 Project Purpose

This project aims to develop a test board, testing methods and testing software for Magnetorquers. Since ADCS has not reached the stage of full electrical integration, the testing tools will facilitate assessments of magnetorquer prototypes and demonstrations of ADCS algorithms in future projects.

This project will also help me build foundational knowledge in PCB design, embedded software, and the operation of magnetorquers, equipping me with the background to tackle more complex technical tasks in ADCS.

1.2 Project Deliverables

The main deliverables for this project are:

- **Magnetorquer Testing Module**
    - PCB
        - Capable of driving controlled current to 3 magnetorquers and obtain feedback (current/voltage measurements)
        - Integrates ESP32 microcontroller, IMU for M-fields and angular rates
        - Independent battery source (battery) to allow free motion
    - Air-coil and Magnetorquer rods connected to PCB.
    - Module able to fit into the prototype CubeSat structure
- **Characterising Magnetorquers using the test board.**
    - Driver Software, Graphing Program, Methods to obtain Characteristic Curves (e.g. Magnetic Dipole Moment vs Voltage)

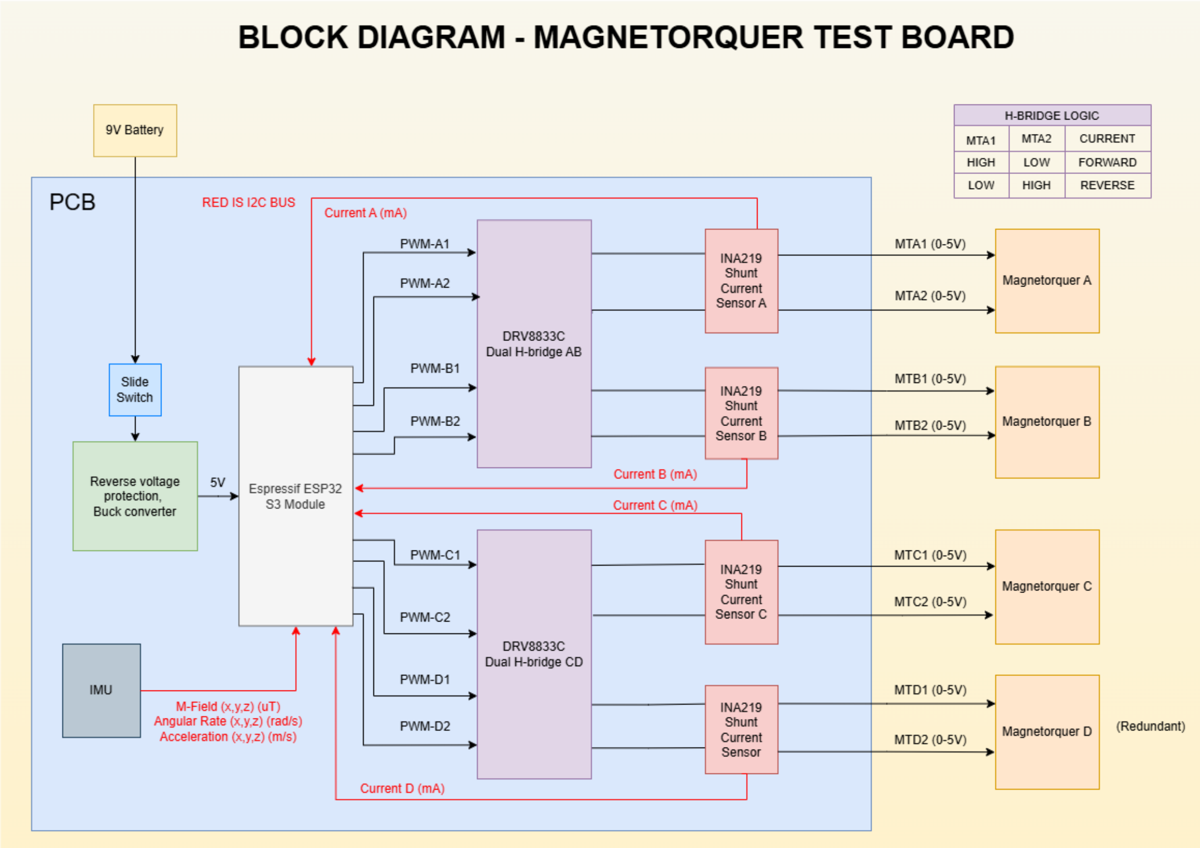
1.3 Project Timeline

I didn't follow a strict timeline, but this is an overview of what I ended up doing each month (backtracked from my Logbook). Project took ~4 months or ~70 hours to complete.

September 2024	Research, planned project, learnt how to use software
October 2024	Component selection, Schematic and PCB Design
November 2024	PCB Revisions, Mechanical Design, Detumbling research and simulation, simple air-coil test, researched testing methods to characterise magnetorquers.
December 2024	Soldering PCB, manufacturing components (Magnetorquers, 3D printed parts), testing PCB and making fixes for revision 2, coded Driver software.
January 2025	Wrote graphing program, Soldered PCB rev2, integrated all components, debugging, tested and characterised magnetorquers, polished up documentation.

2.0 Magnetorquer Test Board

2.1 Test Board Design Overview



Overview

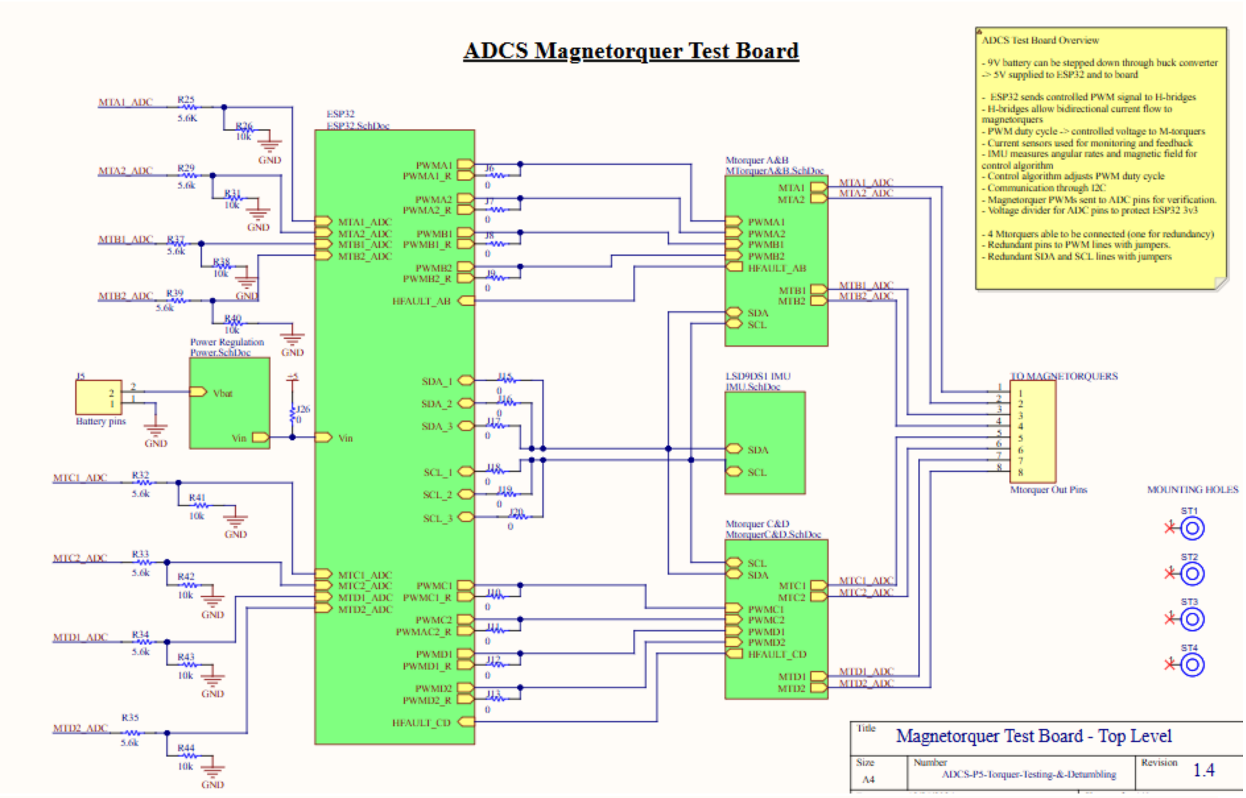
- PCB is powered by a 9V battery.
- 9V gets stepped down to 5V using a buck converter. 5V supplied to the power plane and to the ESP32.
- ESP32-S3 Module is directly connected to the PCB via headers.
- ESP32 sends PWM signals to dual H-bridge IC.
- H-bridge controls magnitude of current through magnetorquers by varying the duty cycle of the PWM signals (average voltage) being sent. Direction changed with high/low voltages to pins (see truth table on diagram).
- 4 magnetorquers can be connected through wires connected to the holes at the edge of the board (one redundant).
- The controlled current through the magnetorquers generates a proportional magnetic field.
- This current is measured by shunt resistor current sensors, which sends current measurement to the ESP32 over I2C.
- The current measurements are used for monitoring and feedback control.
- 9 axis IMU breakout board is connected via headers.
- IMU sends magnetic field, angular velocity and acceleration to the ESP32 over I2C. Used for monitoring and feedback.

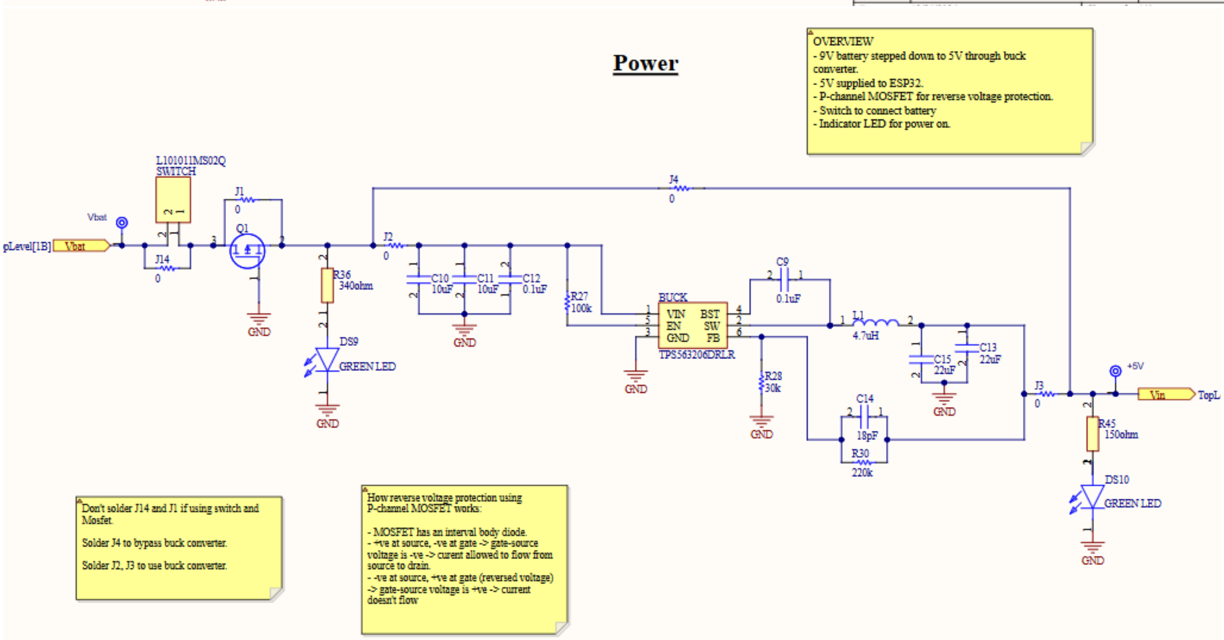
2.2 Schematic and PCB

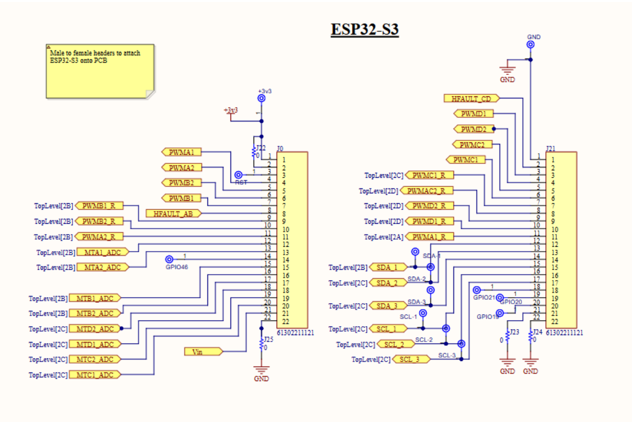
The schematic (annotated with notes) and PCB for the Magnetorquer Test Board can be found on the PAST Altium Workspace. Hierarchal schematic design is used.

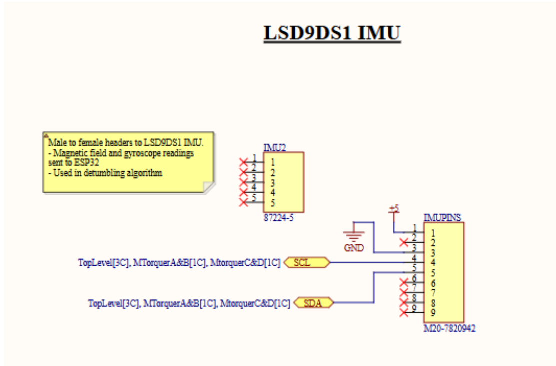
PDF can also be found on SharePoint: [magnetorquer testboard schematic.pdf]

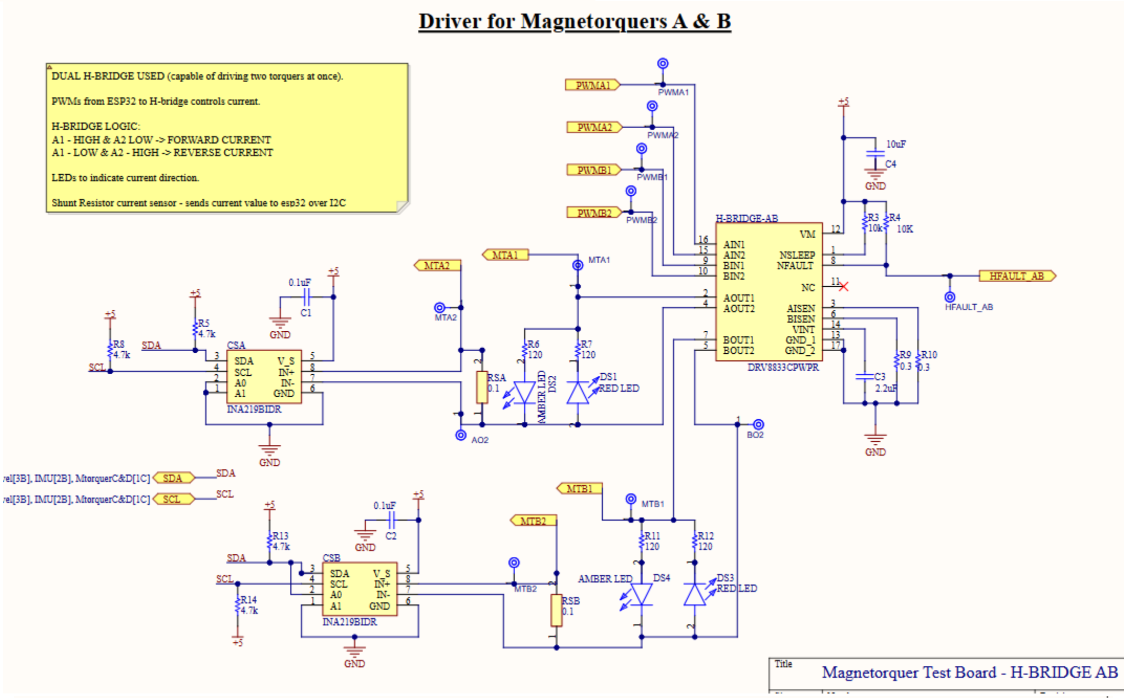
Schematic:

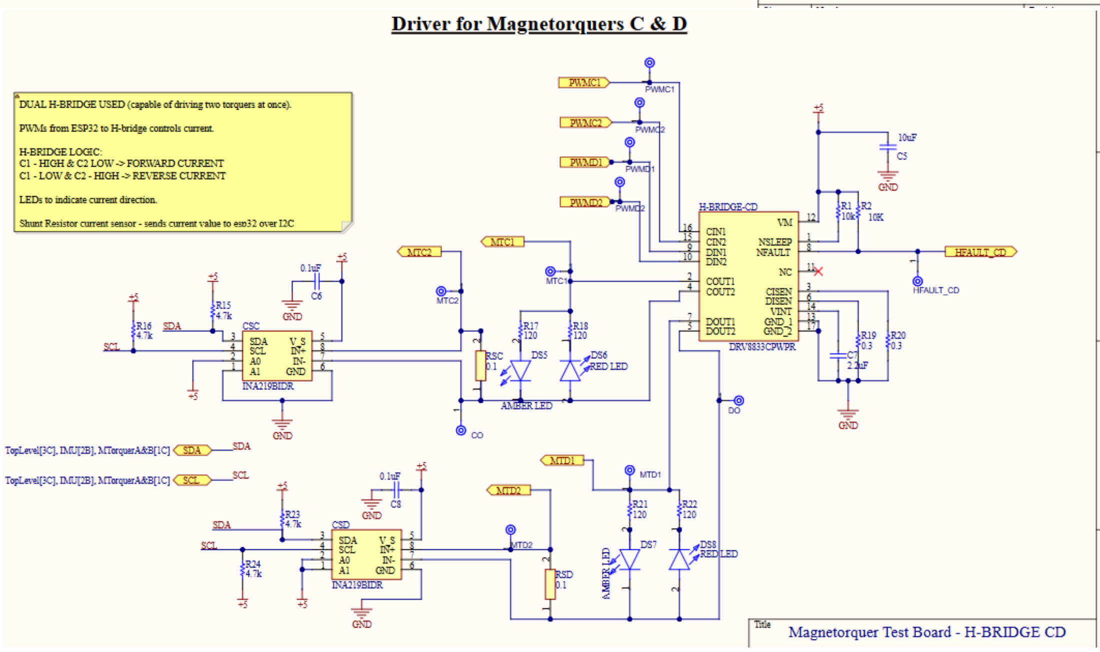




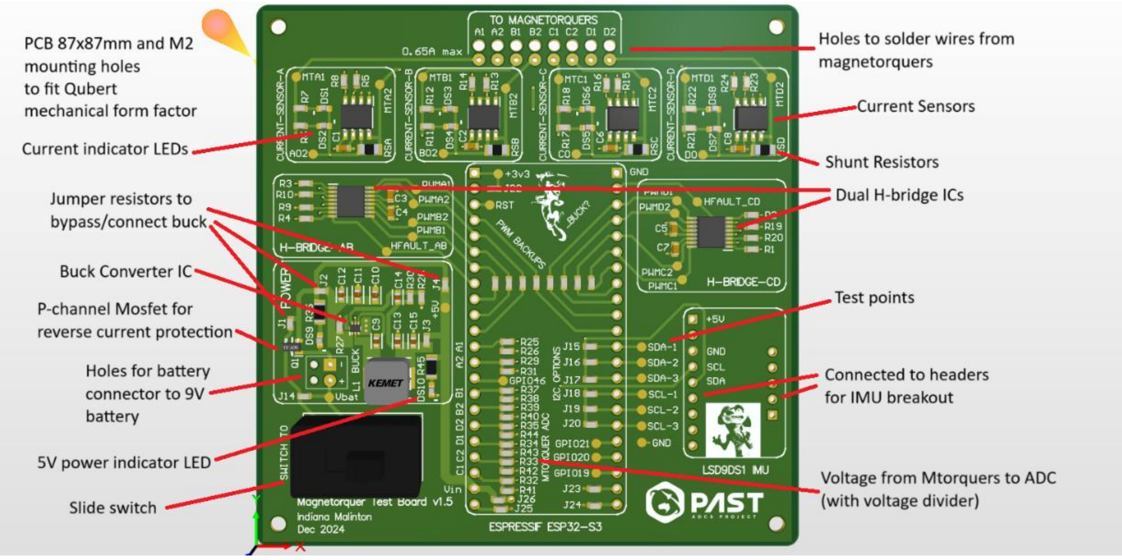

ESP32-S3
Male to female headers to attach ESP32-S3 onto PCB

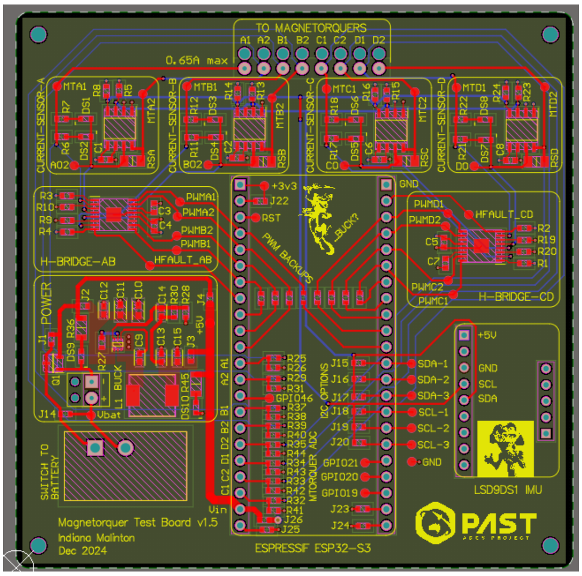

LSD9DS1 IMU
Male to female headers to LSD9DS1 IMU.
- Magnetic field and gyroscope readings sent to ESP32
- Used in detumbling algorithm
IMUPINS
M20-7820942


Driver for Magnetorquers A & B
DUAL H-BRIDGE USED (capable of driving two torquers at once).
PWMs from ESP32 to H-bridge controls current.
H-BRIDGE LOGIC:
A1 - HIGH & A2 LOW -> FORWARD CURRENT
A1 - LOW & A2 - HIGH -> REVERSE CURRENT
LEDs to indicate current direction.
Shunt Resistor current sensor - sends current value to esp32 over I2C
H-BRIDGE-AB
DRV8833CPWPR
INA219BIDR
Title
Magnetorquer Test Board - H-BRIDGE AB


Driver for Magnetorquers C & D
DUAL H-BRIDGE USED (capable of driving two torquers at once).
PWMs from ESP32 to H-bridge controls current.
H-BRIDGE LOGIC:
C1 - HIGH & C2 LOW -> FORWARD CURRENT
C1 - LOW & C2 - HIGH -> REVERSE CURRENT
LEDs to indicate current direction.
Shunt Resistor current sensor - sends current value to esp32 over I2C
H-BRIDGE-CD
DRV8833CPWPR
INA219BIDR
Title
Magnetorquer Test Board - H-BRIDGE CD

PCB:





- Followed Mechanical Form Factor (87x87mm, 2.4mm mounting holes)
- 5mm border for air coil underneath (no pins in the way).
- 4-layer board, 5V & GND plane
- Current sensors on SDA & SCL lines on the back to minimize interference.
- Thermal relief pours on buck converter.
- Exposed pads on extra GPIO pins in case they need to be used.

2.3 Explanation of Components & Design Choices

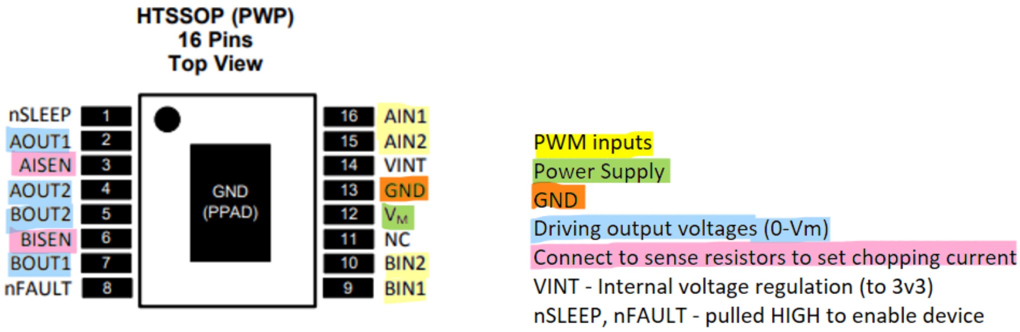
5V power plane

- Originally 3v3 power was implemented, but after initial tests of the air coil, the torquers were found to be relatively weak.
- 3v3 was changed to 5V to enable higher currents through magnetorquers, and stronger magnetic fields to be produced.

Dual H-bridge IC - DRV8833C Dual H-Bridge Motor Driver

Datasheet: https://www.ti.com/lit/ds/symlink/drv8833c.pdf?ts=1735116023867&ref_url=https%253A%252F%252Fwww.google.com%252F

- H-bridges enables the control of the direction and magnitude of current through the magnetorquers – achieved through varying the duty cycle of a PWM signal from MCU and using pin logic (see truth table).
- Contains two full H-bridges in one IC -> able to drive 2 torquers per IC
- Simple PWM interface, built-in current protection.
- Max current 1A per bridge
- Max current able to flow through the magnetorquers is 0.5A due to the current capacity of 30-gauge wire.
- Wide voltage input range 2.7 - 10.8V
- Only fast decay is of interest
- Same H-bridge used in this report: https://courses.grainger.illinois.edu/ece445/getfile.asp?id=16327



H-bridge pin configuration

Table 2. PWM Control of Motor Speed

xIN1	xIN2	FUNCTION
PWM	0	Forward PWM, fast decay
1	PWM	Forward PWM, slow decay
0	PWM	Reverse PWM, fast decay
PWM	1	Reverse PWM, slow decay

H-bridge truth table for controlling current direction

Current Sensors - INA219 Zerø-Drift, Bidirectional Current/Power Monitor With I2C Interface

Datasheet:
https://www.ti.com/lit/ds/symlink/ina219.pdf?ts=1735042442369&ref_url=https%253A%252F%252Fwww.google.com%252F

- Shunt Current Sensor – operates by measuring the differential voltage across the shunt (low value) resistor, then converting to current.

- Bi-directional current, high accuracy and simple to interface with the ESP32 using I2C.
- Shunt resistor current sensors over hall-effect sensors because greatly less affected by electromagnetic interference.
- 16 configurable slave addresses using A1 and A0. Addresses used for each sensor are highlighted below, with hex codes given.

Table 1. INA219 Address Pins and Slave Addresses

A1	A0	SLAVE ADDRESS
GND	GND	1000000
GND	V_{S+}	1000001
GND	SDA	1000010
GND	SCL	1000011
V_{S+}	GND	1000100
V_{S+}	V_{S+}	1000101
V_{S+}	SDA	1000110
V_{S+}	SCL	1000111
SDA	GND	1001000
SDA	V_{S+}	1001001
SDA	SDA	1001010
SDA	SCL	1001011
SCL	GND	1001100
SCL	V_{S+}	1001101
SCL	SDA	1001110
SCL	SCL	1001111

Current Sensor A - 0x40
Current Sensor C - 0x41
Current Sensor B - 0x42
Current Sensor D - 0x45

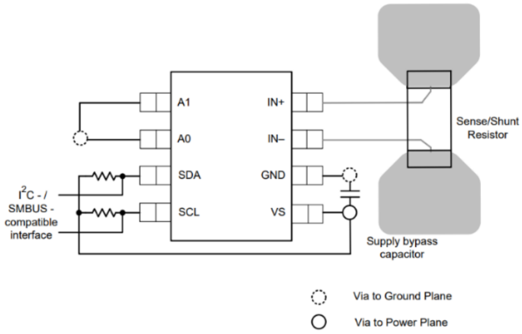


Figure 30. Recommended Layout

To calculate value of shunt resistor:

$$R_{SHUNT} = V_{SHUNT}/I_{max}$$

Vshunt is the maximum differential voltage in the current sensor's range – for this sensor it is 0.1V. Imax is the maximum current the sensor is expected to detect. For my application, Imax = 0.5A -> Rshunt = 0.2 ohms.

LSD9S1 Adafruit 9DOF IMU breakout

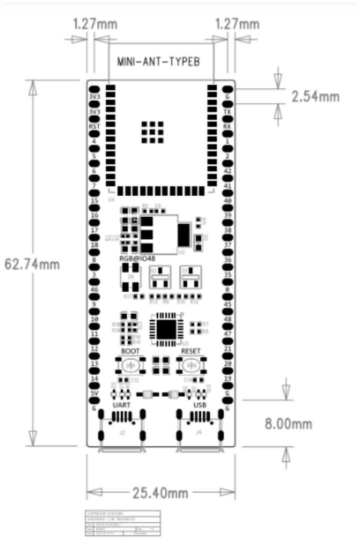
Datasheet: https://cdn-learn.adafruit.com/downloads/pdf/adafruit-lsm9ds1-accelerometer-plus-gyro-plus-magnetometer-9-dof-breakout.pdf

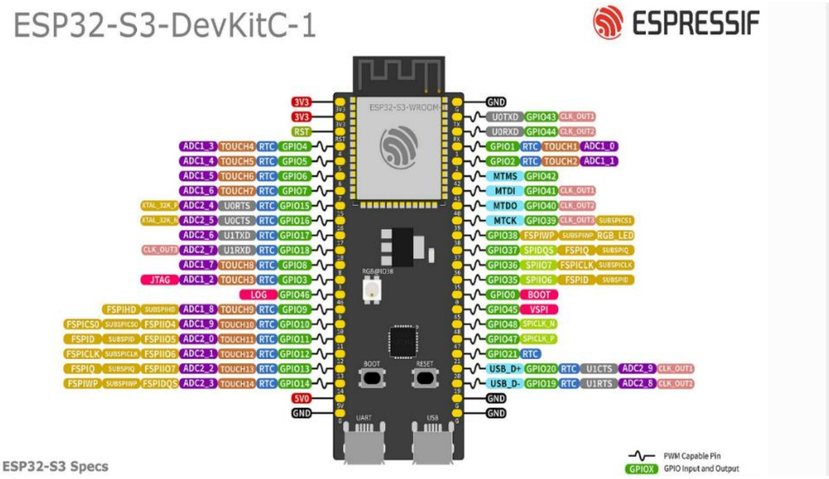
- Adafruit breakout board houses LSD9DS1 chip and power regulation.
- Magnetometer, Gyroscope, Accelerometer and Temperature Sensor is contained in the chip.
- Already had this IMU breakout for testing purposes.
- Repurposed for the test board to save money and time, and reduce points of failure (since I already knew it worked).

Espressif ESP32-S3 Module

Datasheet: **Espressif ESP32**

- ESP32 has built in Wi-Fi and Bluetooth capabilities -> enables test board to move freely when performing detumbling tests in the future.
- Simple Arduino Interface.
- Can be repurposed for prototyping in other projects





ESP32-S3 Specs

Buck Converter

Datasheet:
https://www.ti.com/lit/ds/symlink/tps563203.pdf?ts=1736862587379&ref_url=https%253A%252F%252Fwww.google.com%252F

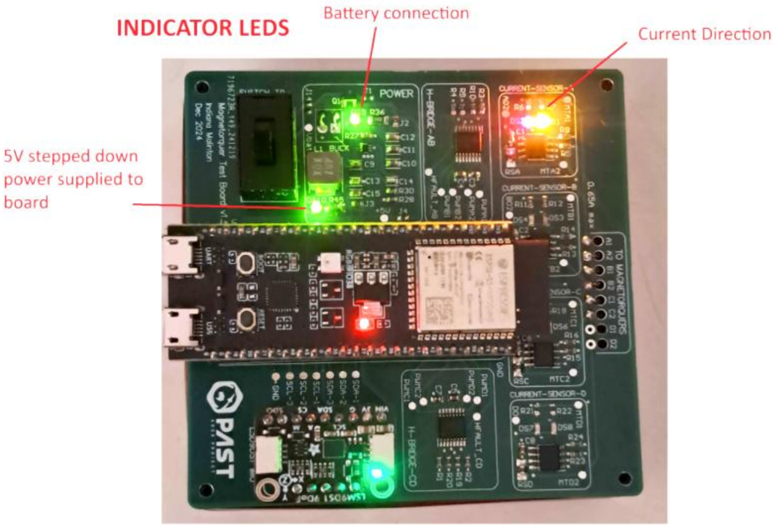
- Input voltage range: 4.2V to 17V, Output voltage range: 0.6V to 7V
- 3A max current – more than enough for 3 magnetorquers.
- Cheap - $0.41 per buck.
- Output voltage is configurable with component values
- Followed layout guidelines of datasheet for routing.

Table 7-2. Recommended Component Values

OUTPUT VOLTAGE (V)	R1 (kΩ)	R2 (kΩ)	Min L(uH)	TYP L (uH)	Max L(uH)	Min Cout(uF)	Typ Cout(uF)	Max Cout(uF)	Typ CFF(pF)
0.8	3.33	10.0	1.2	1.5	3.3	22	66	110	-
1.05	7.5	10.0	1.2	2.2	3.3	22	44	110	-
2.5	95.0	30.0	2.2	3.3	4.7	22	44	110	10
3.3	135.0	30.0	3.3	4.7	6.8	22	44	110	18
5	220.0	30.0	3.3	4.7	6.8	22	44	110	18
7	320.0	30.0	3.3	4.7	6.8	22	44	110	18

Indication LEDs

- Indication LEDs are good practice for easy troubleshooting. They indicate that a voltage exists on a signal line.
- I used LEDs for: battery source power, stepped down (5V) power, two LEDs for each magnetorquers to indicate voltage and current direction.



*The current direction LEDs don't work. They both turn on. I think this is due to some sort of back current. Not too big of a deal in terms of the board's functionality.

To calculate LED resistor values, use:

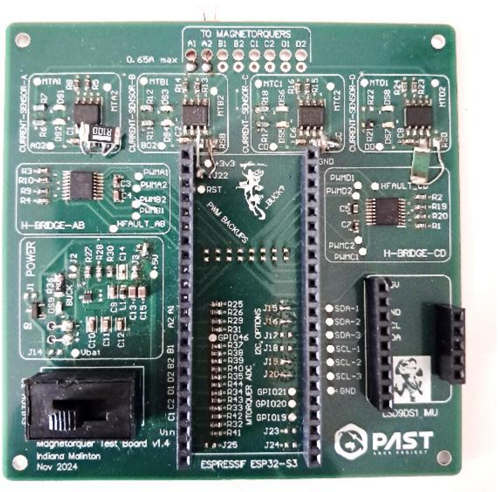
$$R_{LED} = \frac{V_{supp} - V_f}{I_{LED}}$$

Redundant Components/Connections

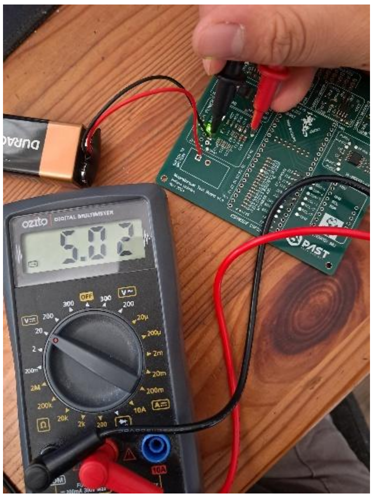
- 2 dual H-bridges – capable of driving 4 magnetorquers. In future, I will use 3 dual H-bridges so each torquer has a redundant H-bridge. Did not do it for this test board due to space constraints on the board.
- Extra current sensor.
- Jumper resistors to bypass components power regulation.
- Redundant connections to multiple ESP32 pins.

2.4 Soldering PCBs

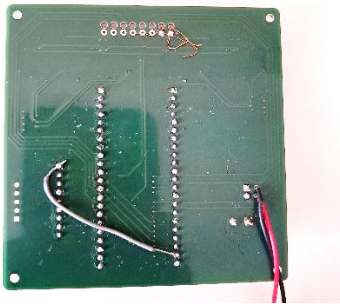
2.4.1 PCB Revision 1



- The PCB had some issues, but I was still able to make it functional!
- Two rev1 boards were soldered.
- This was also my first time soldering with pick and place.

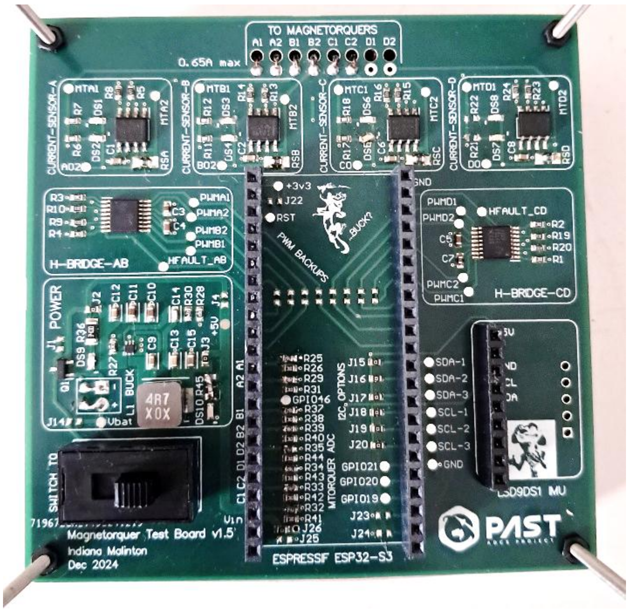


- Buck did not work on one board but worked on the other (probably some buck issue).
- Did not order the components with the correct footprint for shunt resistors, some caps.
- Inductor used in buck circuit did not meet power requirements. For testing, I powered the board through powering the ESP32 with a USB cable.



- Power was not connected to the power plane (got confused when I changed from 3v3 to 5V).
- I was able to connect it with a wire on the back to make it work.

2.4.2 PCB Revision 2



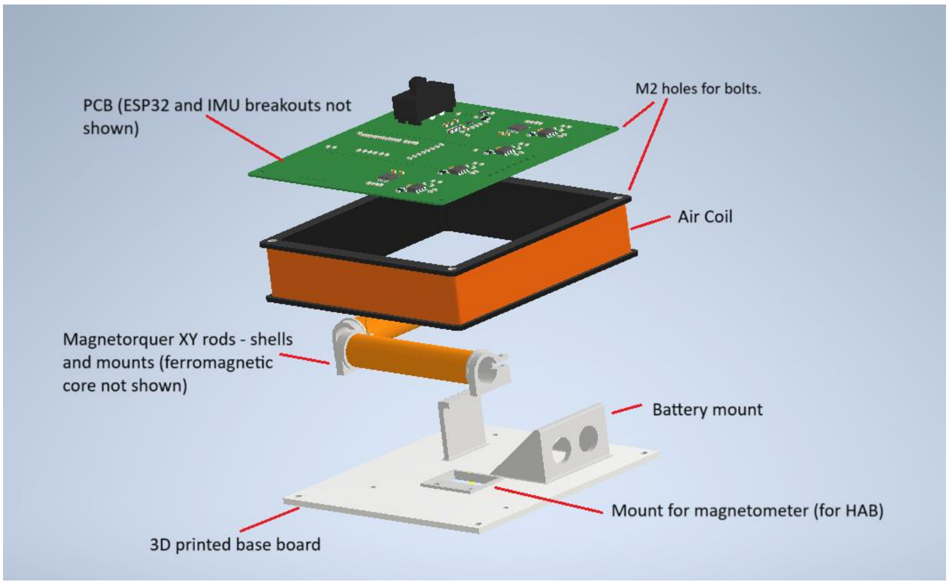
- Main change was modifying the routing of the power circuit to accommodate for the large power inductor.
- Fixed 5V power to the power plane
- Buck converter worked on the first soldered board.
- This time I soldered with 0ohm jumpers instead of bridging which came out a lot neater.
- Only 1 header for the IMU connected, since the right-most is in the way of the air coil underneath.
- Everything worked the first time!

3.0 Structural Design

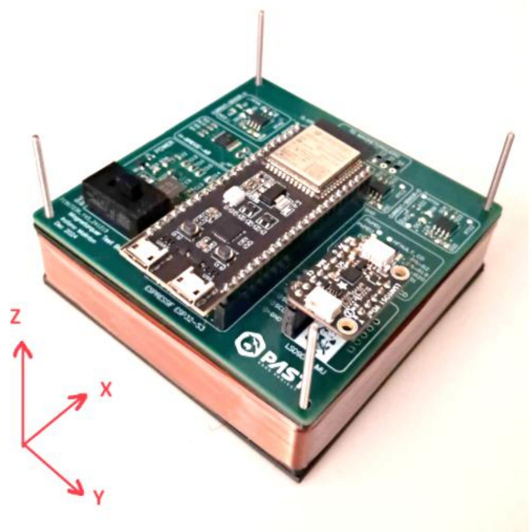
3.1 CAD Model

The 'Magnetorquer Testing Module' consists of the Test board (with ESP32 & IMU attached), air coil, Magnetorquer rods, 9V battery, stacked on top of each other. CAD found on the Project's SharePoint Folder and AutoDesk Drive.

- Designed to fit in an 87x87mm form factor (to fit in Qubert CubeSat Structure)
- PCB is placed on top for ease of monitoring and because magnetorquers cannot fit directly on the PCB.



3.2 Assembly

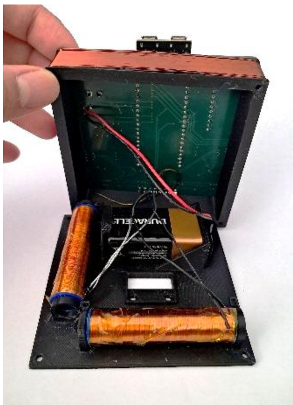
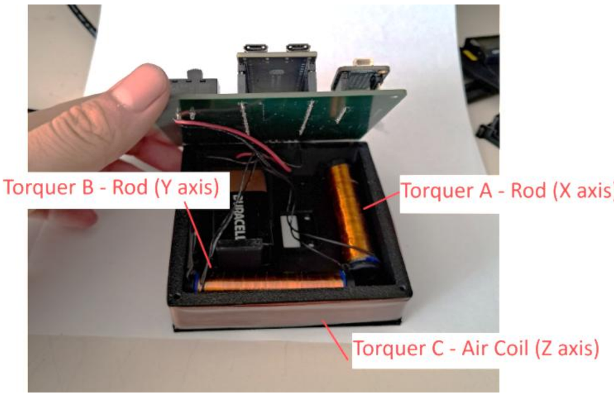


Magnetorquer Module Assembled

- 3D printed air coil frame, base board, rod mounts + shell with PLA.
- M2X50 bolts & M2 nuts – extra length to attach to HAB.
- Wires connected to board from underneath
- NiZn ferrite cores are used in the magnetorquer rods
- IMU is positioned closest to magnetorquer rods at their intersection
- Combined mass of module ~260g
- Total Height (+/- 0.2mm)
  - With ESP32 fully in = 39.6mm
  - Boards & torquers only = 23.6mm
- Length x Width = 87x87mm





Inside Module

- Copper wires insulated with 0.5mm shrink tubing so less prone to breaking & short circuiting.
- Wires colour coded
    - Black = Torquer A
    - Silver = Torquer B
    - Gold = Air coil C

Note:

- It can be hard to plug in/pull out the ESP32 to the headers, so do it gently to prevent bending the pins. A good method is to get your finger under the esp32 and slowly pull upwards.
- Make sure IMU is positioned at intersection of torque rods, since this affects sensor readings.

4.0 Driver Software & Data Analysis

4.1 Driver Code

The software is functional, but it could be improved to be more user-friendly. E.g. User input to turn on which magnetorquer instead of commenting lines. I spent a while trying to manually calibrate the current sensors myself but never got it to scale properly. I ended up using an AdaFruit library.

- Arduino Sketch found on the PAST GitHub and SharePoint.
- Dependencies:
    - Espressif ESP32 Boards → select ESP32S3 Dev Module
    - Wire.h
    - AdaFruit LSM9DS1: https://learn.adafruit.com/adafruit-lsm9ds1-accelerometer-plus-gyro-plus-magnetometer-9-dof-breakout/arduino-code
    - Adafruit INA219: https://learn.adafruit.com/adafruit-ina219-current-sensor-breakout/arduino-code

Note: If getting "A fatal error occurred: MD5 of file does not match data in flash!" when uploading sketch → upload when ESP32 is disconnected from the board.

The code varies the voltage to torquers in a triangular wave form, then prints [Timestamp, duty-cycle, magnetic field (x, y, z) (uT), current (mA)] in csv format to serial.

This is the logic:

```
// Update duty cycle
if (increasing) {
  dutyCycle++;
  if (dutyCycle >= 255) {
    increasing = false;
  }
} else {
  dutyCycle--;
  if  (dutyCycle <= 10) {
    increasing = true;
    directionSwitch = !directionSwitch;
  }
}
```

```
  // Adjust PWM duty cycle and direction of current
  if (directionSwitch) {
    ledcWriteChannel(PWM_CHANNEL_A1, dutyCycle); // A1 = HIGH, A2 = LOW
    ledcWriteChannel(PWM_CHANNEL_A2, 0);
  } else {
    ledcWriteChannel(PWM_CHANNEL_A2, dutyCycle); // A1 = LOW, A2 = HIGH
    ledcWriteChannel(PWM_CHANNEL_A1, 0);

  }
}
```

Increasing – boolean to toggle between increasing or decreasing PWM duty cycle.

directionSwitch - boolean to reverse currents by flipping H-bridge inputs when duty cycle reaches 0.

4.2 Graphing Program

I made a simple Python Program to graph data from the Serial.

- Program found on PAST GitHub and the Project's SharePoint Folder
- Dependencies: serial, csv, datetime, matplotlib.pyplot
- Program reads serial data and writes to a csv file. Then reads the csv file and creates 6 subplots.

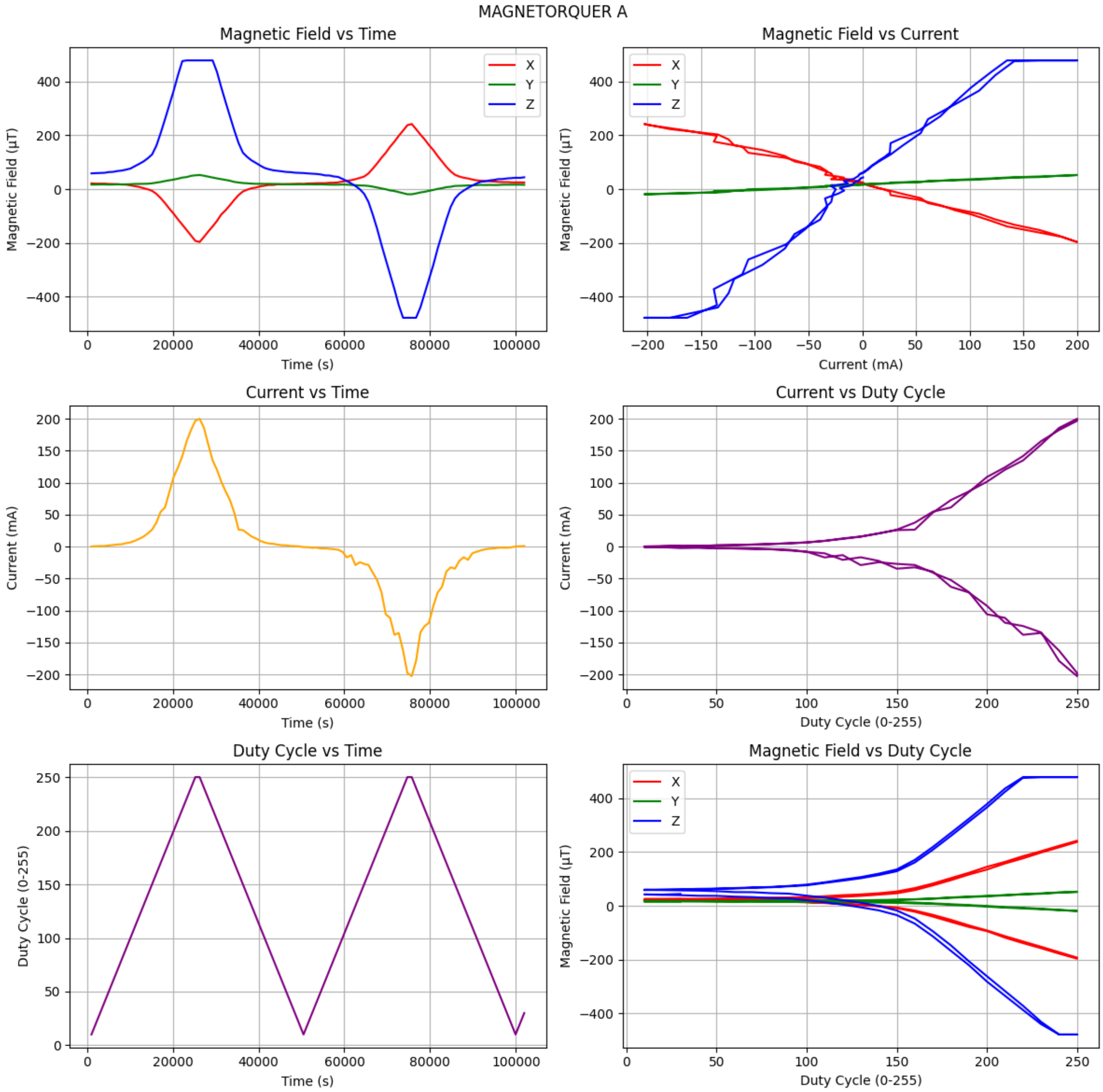
How to use it:

1. Make sure an appropriate sketch is uploaded to the ESP32, which prints data to serial.
2. Anything that could cause interference must be least 20cm away if wanting pure magnetorquer measurements.
3. Connect the ESP32 to your COM port (make sure it is the same in the code, it is COM8 by default) then run the program.

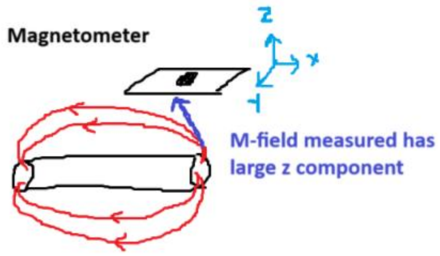
4.3 Graph Results

- I obtained plots for each magnetorquer in the module (in the orientation outlined in Assembly).
- More plots are found on SharePoint: GRAPHS
- I varied voltages in a triangular wave form (0 → 5V → 0 → -5V → 0).
- Found that reverse currents are noisier.
- Position of IMU affected readings.
    - In final board, magnetometer for feedback should be placed on the same level as the torque rods.
    - This does not impact the purpose of the board, since the torquers do not need to be stacked when characterising magnetorquers, and algorithms are interested in relative change rather than values.

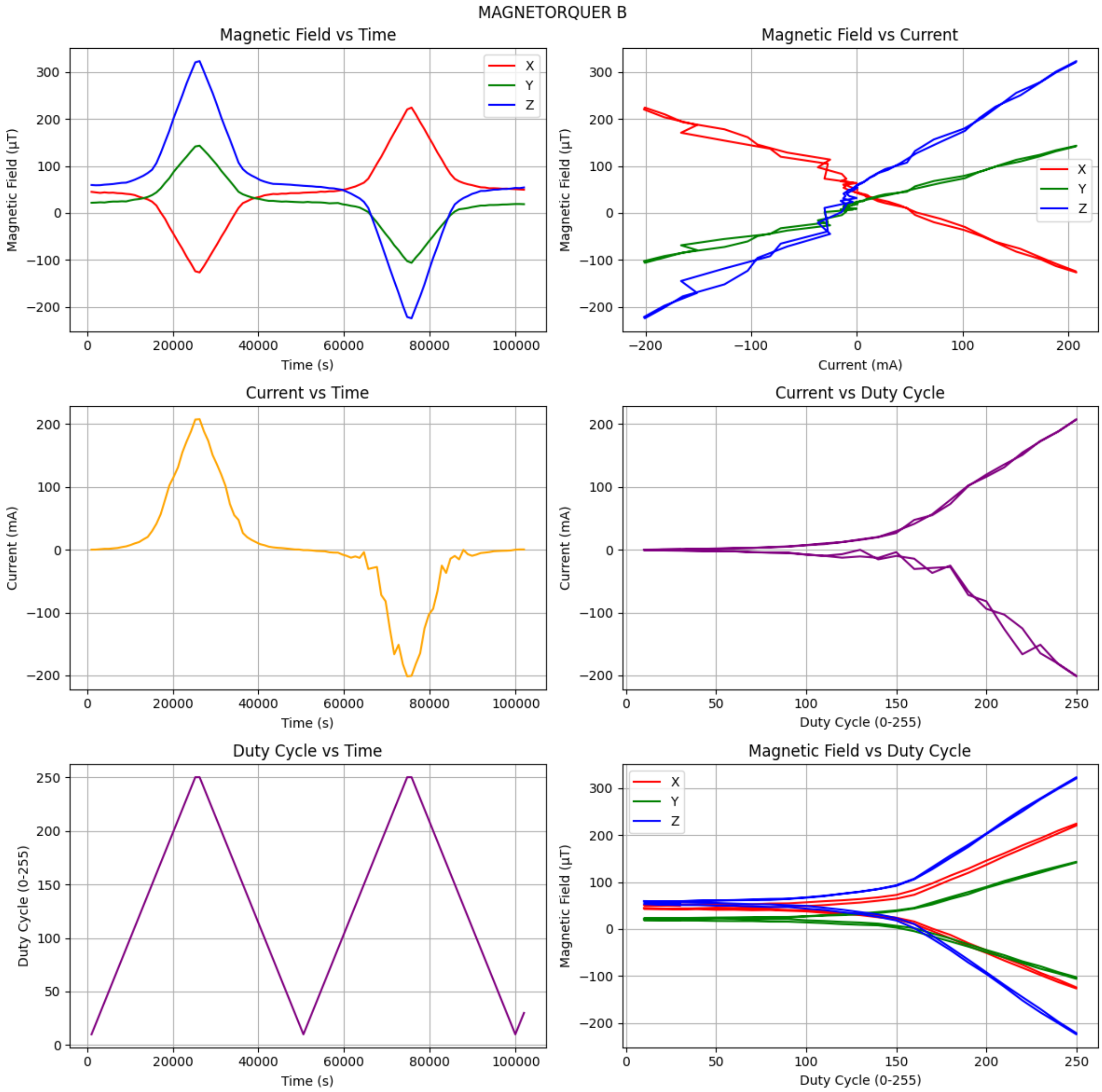
Magnetorquer A (X-axis Rod)



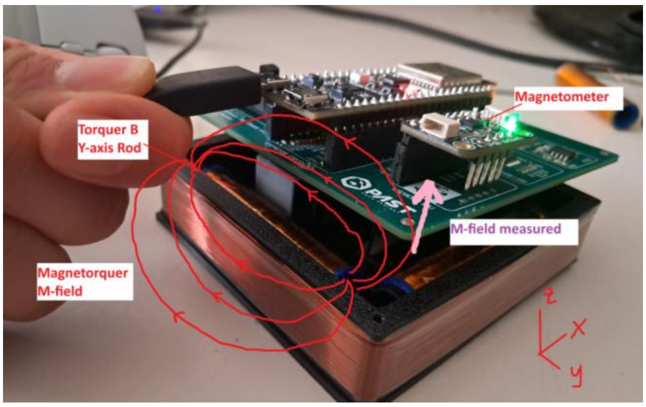
- Even though it's the X-axis rod, the sensor picked up the Z component most.
- This is likely due to the position of IMU.



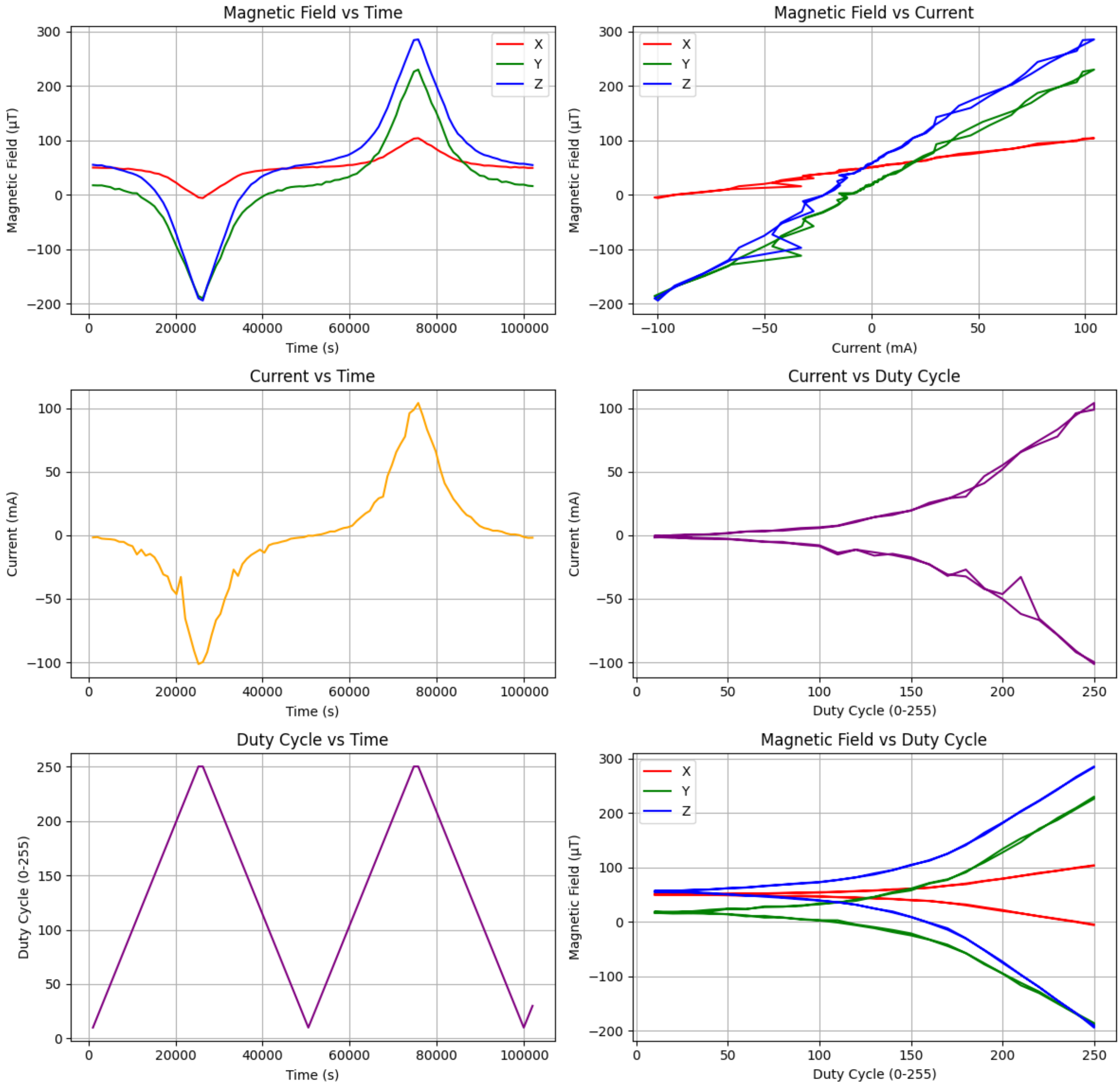
Magnetorquer B (Y-axis Rod)



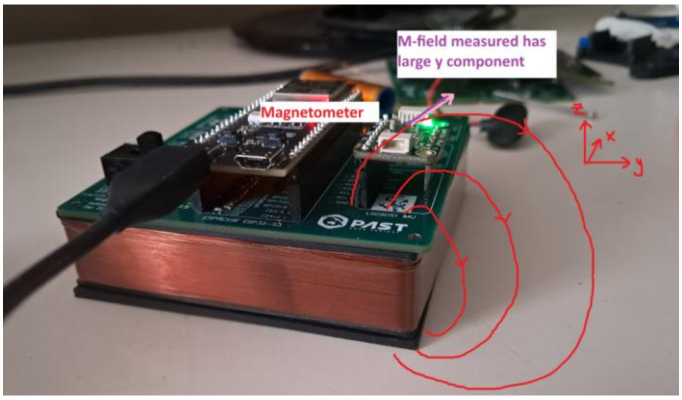
- The position of the IMU affected the readings once again.
- While this is the Y-axis rod, the majority of M-field measured was Z and X components.
- Image below illustrates what I mean by the position of IMU affecting measurements.



Magnetorquer C (Z-axis Air Coil)



- IMU position affected measurements again, as illustrated below.



5.0 Magnetorquer Characterisation

Magnetorquers are typically characterised by their Magnetic Dipole Moment (with units Am^2), which is the object's tendency to align itself with an external magnetic field. Knowing the M-dipole is important for ADCS algorithms, since the torque due to a magnetorquer is the cross product of magnetic dipole moment (μ) with the external magnetic field (B), and to get to a desired orientations → need to apply appropriate torques.

$$\tau = \mu \times B$$

5.1 Method to obtain M-dipole

I referenced this paper for the experimental method to determine magnetic dipole moment: https://www.researchgate.net/publication/242270492_On_Determining_Dipole_Moments_of_a_Magnetic_Torquer_Rod_-_Experiments_and_Discussions

I summarised the method here on SharePoint: Experimental Method to Determine M-dipole of Mtorquers

Basic idea of experiment: Use the equation below to obtain the Magnetic Dipole Moment (M), measured in Am^2, by taking magnetic field readings (Br) at different voltage levels while keeping R and L constant. To do this:

1. Make sure the ESP32 is programmed to vary voltages from (0 → 5V → 0 → -5V → 0).
2. Set up magnetorquers as shown below.
3. Use the graphing program to obtain the magnetic field and voltage data in a csv file.
4. Use Excel for data analysis to obtain the characteristic curve.

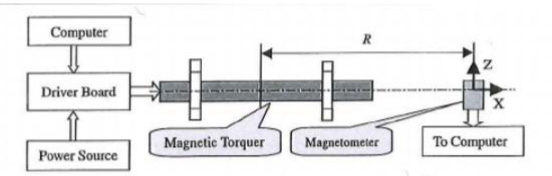


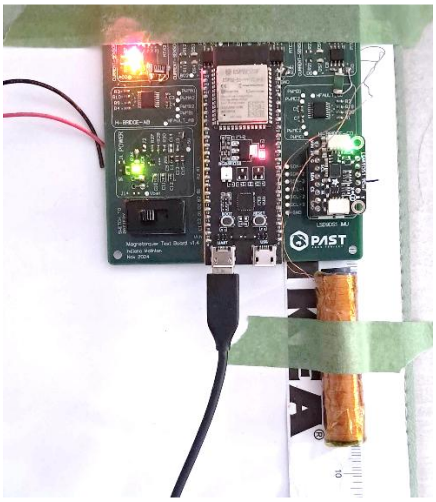
Figure 2. Experimental Setup to Measure Magnetic-Flux Density – Axial Placement of Magnetometer.

If $\theta = 0°$,

$$M = \frac{4\pi}{\mu_0} \cdot \frac{1}{\dfrac{\frac{R}{L} - \frac{1}{2}}{\left(R^2 - RL + \frac{L^2}{4}\right)^{3/2}} - \dfrac{\frac{R}{L} + \frac{1}{2}}{\left(R^2 + RL + \frac{L^2}{4}\right)^{3/2}}} \cdot B_r \quad (8)$$

5.2 Test Set Up

I could have conducted the test more rigorously, but this was more for getting a rough idea of the magnetic dipole moment and demonstrating how to use the method. I'd like to save characterising magnetorquers as a project for other ADCS members in the future, also since the scope of this project was getting quite inflated.

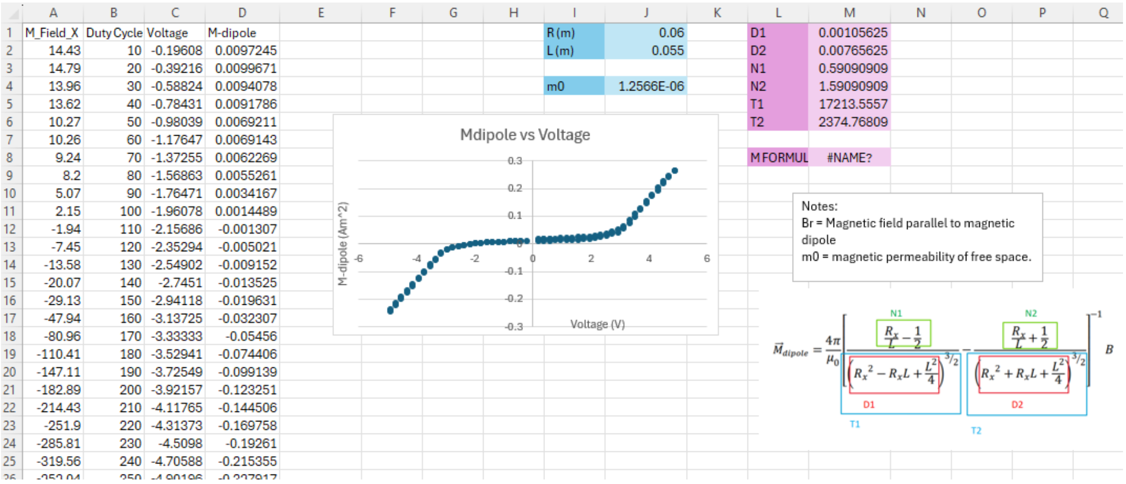


Magnetorquer Rod Set Up

- Tested with NiZn ferrite core
- Taped Magnetorquer Rod r=60mm away from magnetometer (on IMU).
- Aligned Torque Rod Dipole with magnetometer axis.
- No ferromagnetic material within 20cm radius
- Varied voltages from (0 → 5V → 0 → -5V → 0)
- Errors from my IKEA measuring tape, magnetometer being slightly elevated from magnetorquer rod, magnetometer calibration/noise, nearby signals on board.

5.3 M-dipole Data Analysis & Results

Spreadsheet for Data Analysis can be found on SharePoint: Mtorquer Data Analysis



M_Field_X	Duty Cycle	Voltage	M-dipole
14.43	10	-0.19608	0.0097245
14.79	20	-0.39216	0.0099671
13.96	30	-0.58824	0.0094078
13.62	40	-0.78431	0.0091786
10.27	50	-0.98039	0.0069211
10.26	60	-1.17647	0.0069143
9.24	70	-1.37255	0.0062269
8.2	80	-1.56863	0.0055261
5.07	90	-1.76471	0.0034167
2.15	100	-1.96078	0.0014489
-1.94	110	-2.15686	-0.001307
-7.45	120	-2.35294	-0.005021
-13.58	130	-2.54902	-0.009152
-20.07	140	-2.7451	-0.013525
-29.13	150	-2.94118	-0.019631
-47.94	160	-3.13725	-0.032307
-80.96	170	-3.33333	-0.05456
-110.41	180	-3.52941	-0.074406
-147.11	190	-3.72549	-0.099139
-182.89	200	-3.92157	-0.123251
-214.43	210	-4.11765	-0.144506
-251.9	220	-4.31373	-0.169758
-285.81	230	-4.5098	-0.19261
-319.56	240	-4.70588	-0.215355

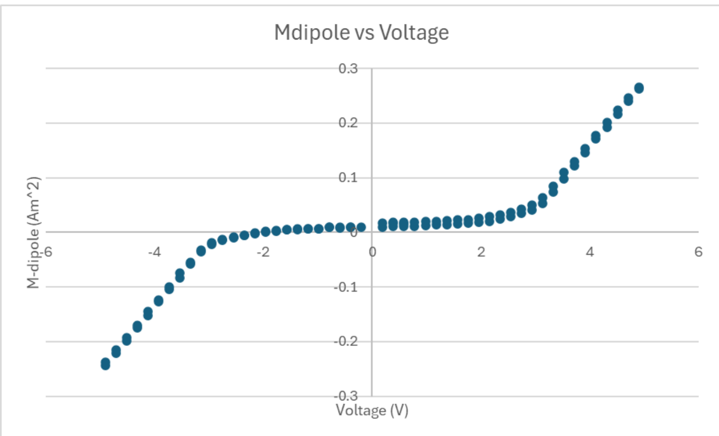
R (m)	0.06
L (m)	0.055
m0	1.2566E-06

D1	0.00105625
D2	0.00765625
N1	0.59090909
N2	1.59090909
T1	17213.5557
T2	2374.76809
M FORMUL	#NAME?

Notes:
Br = Magnetic field parallel to magnetic dipole
m0 = magnetic permeability of free space.

$$\vec{M}_{dipole} = \frac{4\pi}{\mu_0}\left[\underbrace{\frac{\overbrace{\frac{R_x}{L}-\frac{1}{2}}^{N1}}{\underbrace{\left(R_x^2 - R_xL + \frac{L^2}{4}\right)}_{D1}^{3/2}}}_{T1} - \underbrace{\frac{\overbrace{\frac{R_x}{L}+\frac{1}{2}}^{N2}}{\underbrace{\left(R_x^2 + R_xL + \frac{L^2}{4}\right)}_{D2}^{3/2}}}_{T2}\right]^{-1} B$$

Torque Rod



4 M-dipole range (Am^2): (-0.24, 0.26)
5 Linear after 3V

Future work would involve characterising the air coil. I did not do it myself because I already soldered all three magnetorquers to the board and it is a pain to desolder the rods to isolate the air coil. Oops.

5.4 Other Magnetorquer Specifications

The manufacturing and design of magnetorquers is covered by a separate project, but I thought I'd also include some rough specs of the magnetorquer prototypes I tested with, since that separate project is not finished yet.

Magnetorquer Rods (X & Y axis)	• Resistance 19 – 20 ohms • 4 layers of enamelled copper wire, wound using the BINAR coil winder onto a PLA shell. • NiZn Ferrite core • Max current ~ 0.2A • Max M-dipole ~ 0.26Am^2
Air coil (Z axis)	• Resistance ~41 ohms • Wound by hand (I forgot how many layers I did *cry*), but more than 4. You can find the approximate length of wire used by resistance/resistivity of wire. • 3D printed PLA frame • Max current ~ 0.1A

6.0 Components & Cost Breakdown

All costs recorded came out of PAST's 2024 budget.

Part	Notes	Qty	Cost
Espressif ESP32-S3 Modules	Development boards can be used by other projects as well.	3	$77.72
PCB rev1 + Stencil	Board = $30.68, Shipping = $ 43.58 ☹️	5	$74.26
PCB rev2 + Stencil	Fixed power plane and power inductor. Shipped with other boards.	5	$18.90
H-bridge ICs	-	10	$21.56
Current Sensors	-	16	$57.20
Buck Converters	-	20	$9.81
Slide switches	-	5	$30.42
Socket Headers	To attach ESP32 and IMU. 10x5pin, 10x9pin, 10x22pin		$77.99
Battery snap connectors	-	10	$7.43
RLC components	Surplus ordered for general stock.	-	$115.51
3-D printed parts (magnetorquer frames, module's base)	My own prints.	-	-
9V battery	Just found some batteries lying around.	-	-
M2X50 bolts & nuts	Sourced from my dad's garage.	-	-
IMU breakout	Already procured under a different project.	-	-
		Total	$490.80

The total cost of $490.80 is under the proposed budget estimate of $570.

7.0 Reflection & Lessons Learnt

Overall, this project was super fun! It helped me build skills, fluency and confidence in many technical areas – PCB design, reading datasheets, Arduino programming, magnetorquer analysis, CAD, data analysis, and most importantly how to draw on these various skills to make an integrated system.

I think the project was successful. It met its deliverables within a reasonable timeline, and produced meaningful tools and results that will help PAST develop ADCS for the final CubeSat. A factor for this project's success is that I was lucky enough to get support through Engineering hours meetings, mostly with the PCB design since it was my first time. Other than that, I learnt a lot through reading papers and reports from other universities, watching YouTube videos, a lot of thinking, and a lot of trial and error.

In attempt to continue the cycle of knowledge-sharing, I placed a lot of emphasis on maintaining clear documentation of my project. I hope this document will help future PAST members learn how to use the test board, testing methods and testing software, and give them insight into my overarching design process.

Key Lessons Learnt

- In the first two months, the scope of the project kept changing. I spent too long doing PCB revisions (see section 10.1). It would have been more time efficient to have had a set idea of the PCB from the start.
- Order components of the correct footprint the first time to mitigate some cursed soldering. 0603 is in imperial units. However, even if footprints/connections are wrong, there are still ways to make the board functional.
- Take advantage of tools available to you.
    - I wasted a lot of time trying to calibrate the current sensors manually when I could have simply searched for and used a library from the start.
- Learnt troubleshooting. Examples:
    - No power being supplied to the board → maybe the buck wasn't working? → multi meter tests shows its working → maybe a soldering issue? → everything seems soldered correctly → looking back at PCB routing → realised power wasn't connect to the power plane dammit.
    - Magnetorquer B wasn't turning on → wasn't soldered properly.
    - Air coil wasn't turning on → maybe not soldered properly again? → resolder got it working in one direction but not the other → maybe LEDs are the wrong way round and affecting it? →nah → staring schematic for a while → I give up → next day staring at code → hmm only possible fault is the pin assignment → checks ESP32 pinouts → realised a pin was assigned wrong in the code AGHHH.

8.0 Future Work

More ADCS projects/work can be brought to fruition by using the test board, driver software and graphing program as tools. Future work could include:

Potential Improvements for this Project

- Add more functionality to the graphing program (e.g. make it a GUI, add live monitoring).
- Communicating with the board via Bluetooth/Wifi.
- Verify Voltage across magnetorquer using ADC pins.

HAB Testing with Test Board

- I will need to write the software for HAB testing.
    - Current Idea is to alternate between the 3 magnetorquers, varying duty cycle in a triangular wave form (paused at peak for longer)
- Need to figure out how to pipe data to flight computer or black box.

ADCS Algorithm Testing

- Test board can be used to demonstrate/prove the functionality of algorithms. E.g. detumbling.

Characterisation of Magnetorquers

- Use the test board, testing method and software to characterize future revisions of magnetorquer rods.
- Characteristic curves of magnetorquers can be used in dynamic modelling equations and software.
- Investigating the effects of having all three magnetorquers together (any EMF induced?).

ADCS Integration Board

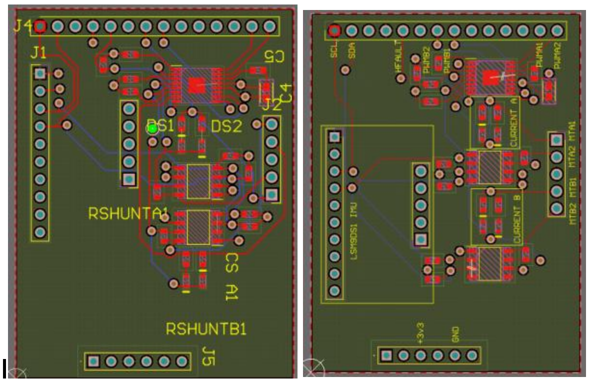
- H-bridge and current sensors can be used in the final ADCS integration board.
- Using a Microcontroller chip and IMU chip instead of breakouts.
- Magnetorquers mounted onto the PCB rather than on a separate board underneath.

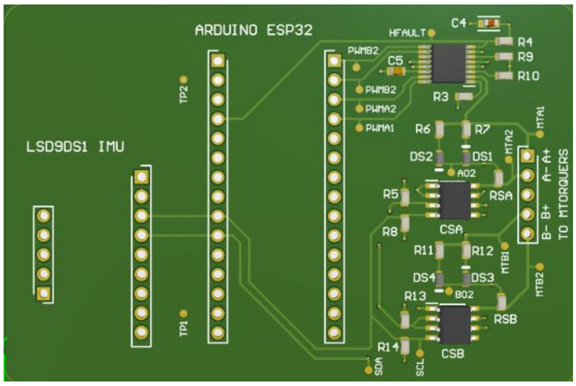
9.0 Project Closeout Approval

Role	Date	Name	Signature
Project Member	24/01/2025	Indiana Malinton	Indi

10.0 Cool Extra Stuff

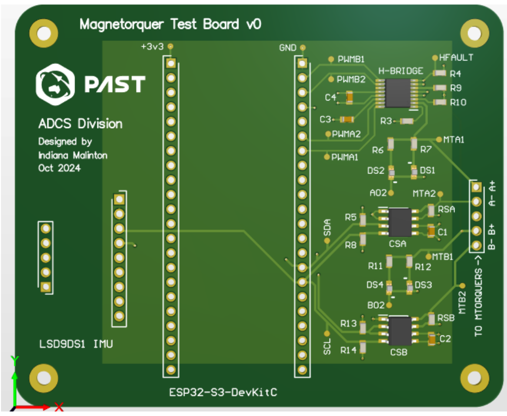
10.1 Previous PCBs



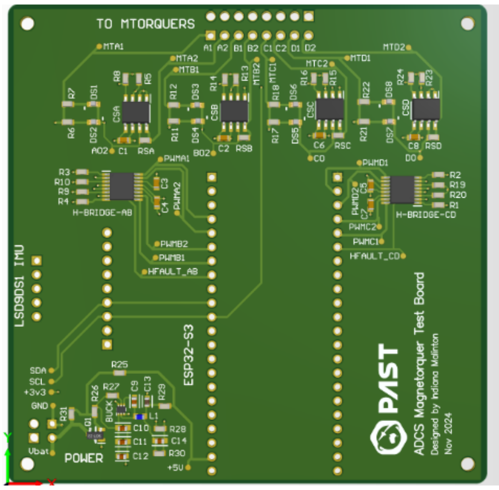


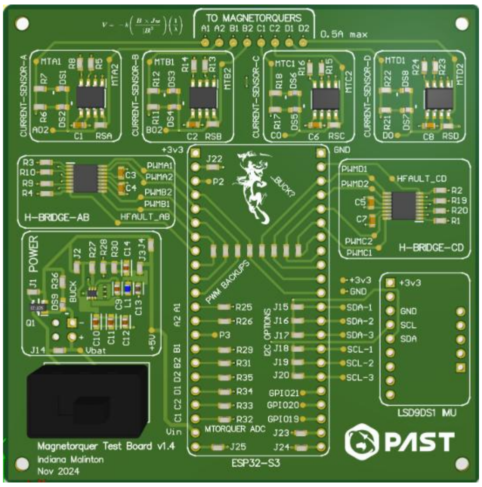
At first it was meant to be a shield for an Arduino UNO, only be able to drive two torquers. I wanted to finish this project quick so I could get onto algorithm stuff.

I then realised that data needed to be communicated wirelessly if I wanted to test detumbling. Aman proposed ESP32 as a solution, so I thought that meant Arduino ESP32.



Turns out it was meant to be an Espressif ESP32 module LOL.



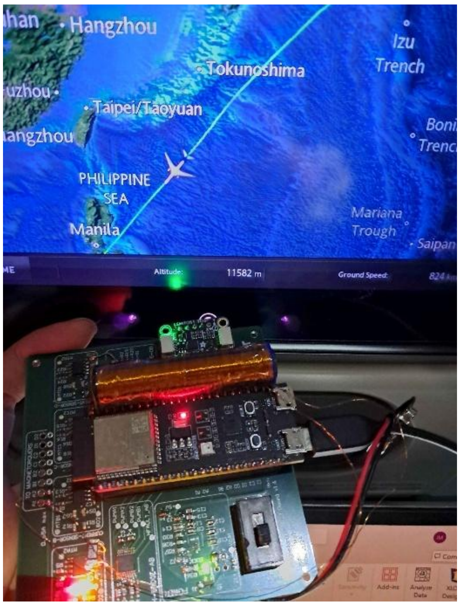


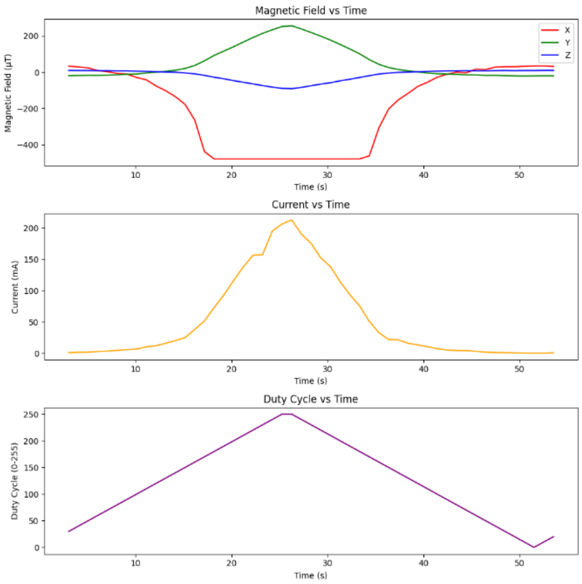
Then I was like wait a minute, I need to power this somehow if we are attaching the ESP32 like this – Aman suggested 9V battery. I figured I'd step the voltage down with a buck converter. I used a jumper system; in case it failed. Nearly convinced to accommodate 6 magnetorquers but I settled with 4 due to space constraints.

Raph luckily spotted I lacked thermal relief pours on the buck routing. I learnt the importance of redundancy and added it everywhere. Added a fat ass switch because I had the room and good for testing. Also spent a bit too long making this board pretty XD.

10.2 Magnetorquer working at 11.6km altitude

Yippee!





10.3 Detumbling Equations & Simulation.

I removed detumbling from the scope of this project to save it for other members. This is some stuff I did:

1. $M = -k \frac{B \times Jw}{|B|^2}$

$$\lambda = (\frac{r}{2W_{res}})(1 + \frac{\mu_r - 1}{1 + (\mu_r - 1)N_d})$$

2. $M = V\lambda$

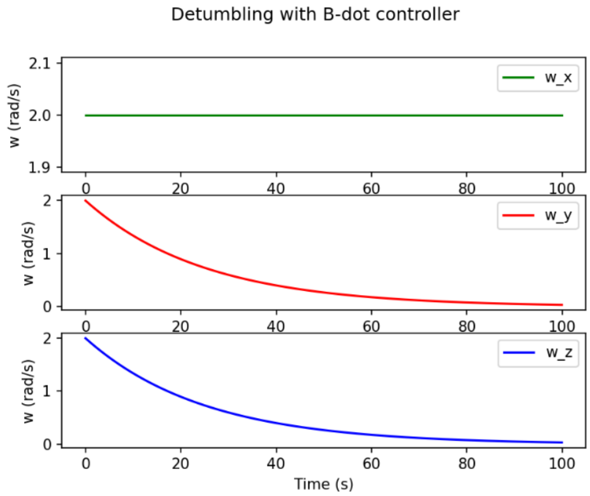
Combining (1) and (2),

$$V = -k(\frac{B \times Jw}{|B|^2})(\frac{1}{\lambda})$$

Where V is a vector of the voltages supplied to each magnetorquer, k is the gain factor to adjust the detumbling rate, B is the magnetic field vector from the magnetometer on the IMU, J is the inertia tensor of the spinning body, w is the angular velocity vector from the gyroscope on board the IMU, and λ is a constant factor from the characteristics of the magnetorquer.

I also simulated the classic B-dot control law (different to above):

The equation is simple enough to simulate using Euler integration.




```python
for i in range(len(tspan)):
    # B-dot control law: magnetic dipole moment of magnetorquers
    B_dot = np.cross(B, w)
    m = -k * B_dot

    # Torque due to magnetic dipole moment
    t = np.cross(m, B)

    # Angular acceleration using dynamic model
    w_dot = np.matmul(Jinv, t - np.cross(w, np.matmul(J, w)))

    # Update angular velocity with time step
    w = w + w_dot * t_step

    # Store values for plotting
    w_xvalues[i] = w[0]
    w_yvalues[i] = w[1]
    w_zvalues[i] = w[2]
```

10.4 Cursed soldering

Don’t try this.